Operational Forecasting at NCEP

A glimpse

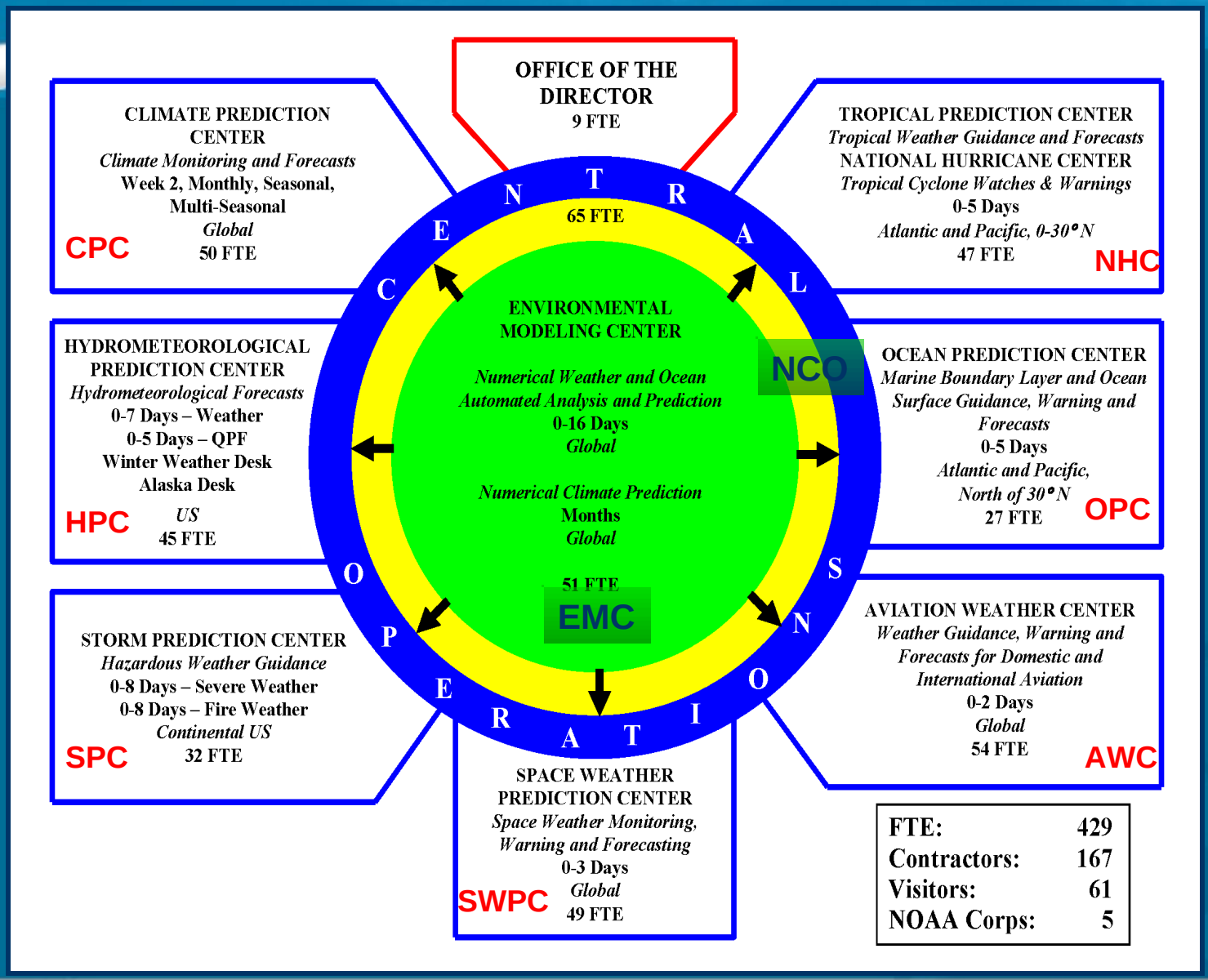
Arun Chawla
SAIC @ Marine Modeling and Analysis Branch
NOAA / NWS / NCEP / EMC

Arun.Chawla@NOAA.gov

Quality of an Operational Model

- What makes an operational model good ?
  1. RELIABILITY
  2. RELIABILITY
  3. RELIABILITY
  4. On time deliverable
  5. On time deliverable
  6. On time deliverable
  7. Accurate results

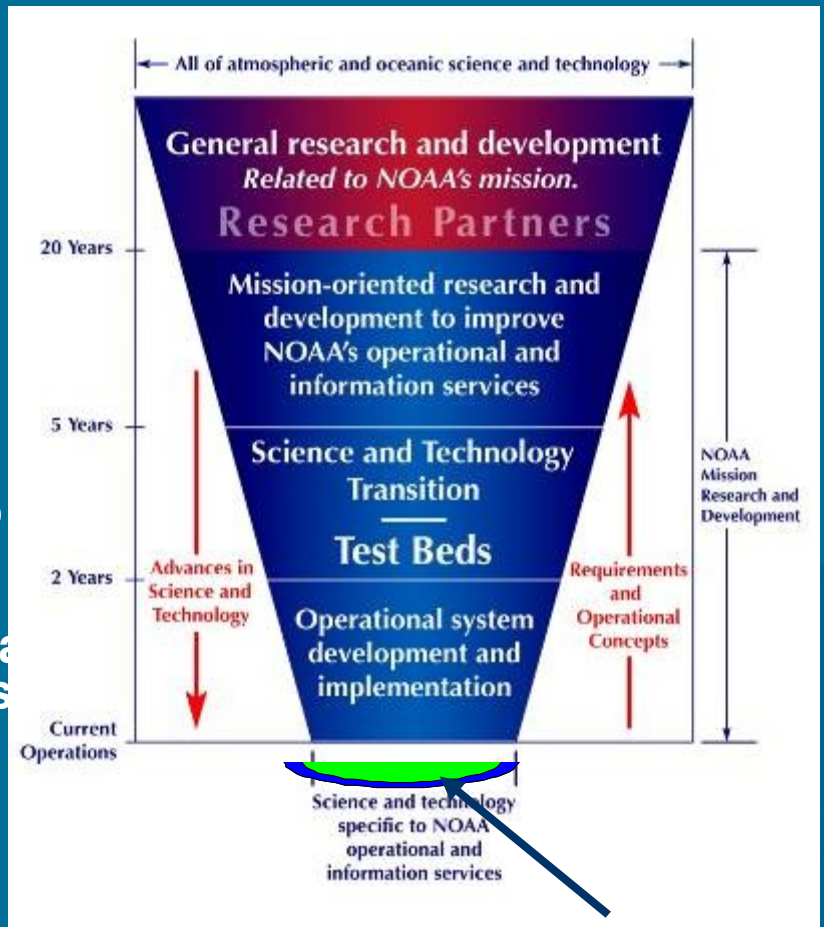
National Centers for Environmental Prediction



The EMC Mission…..

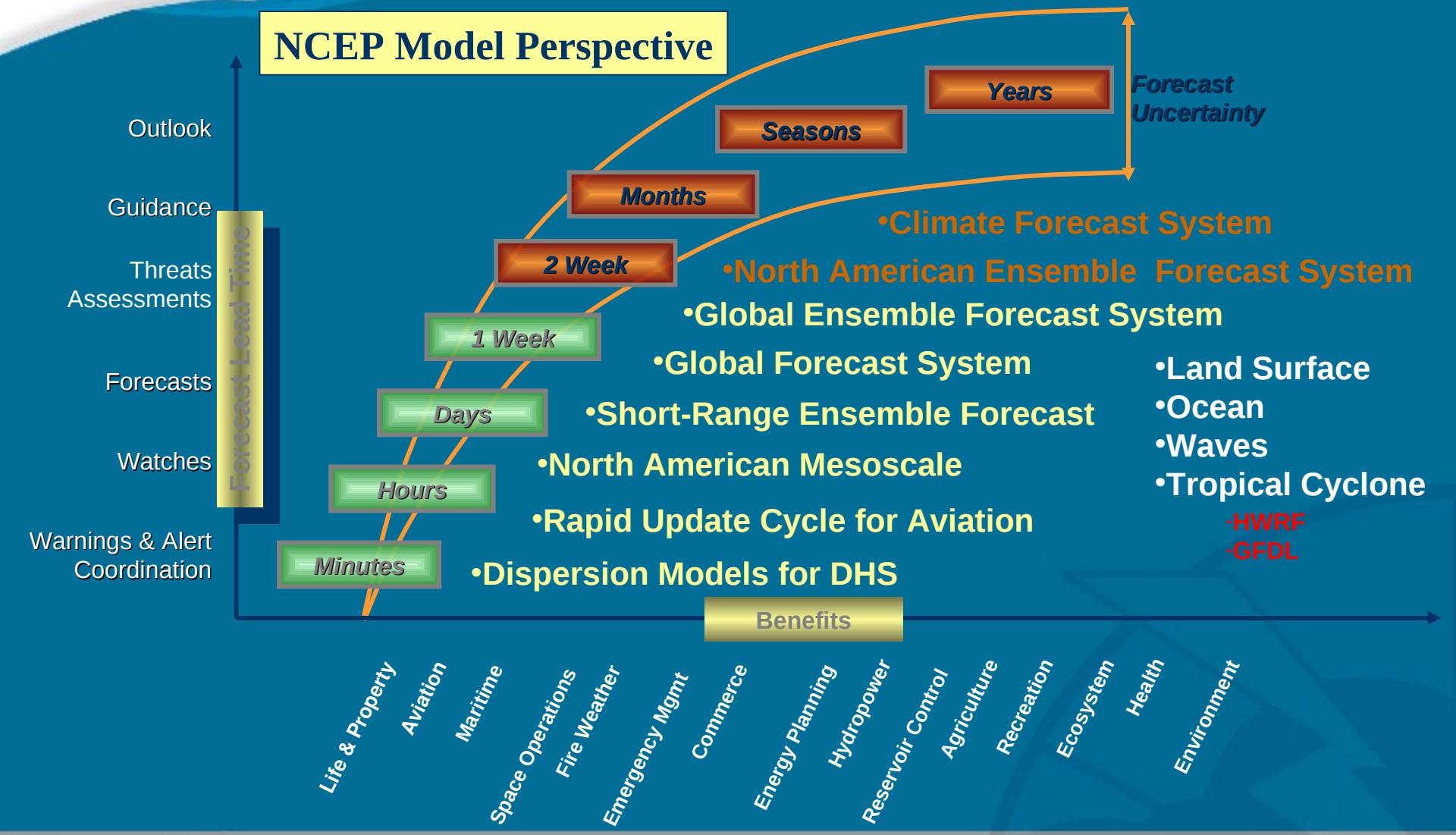
In response to operational requirements:

- **Enhance** numerical guidance
  - Test and improve NCEP's numerical forecast model systems via
    - Scientific upgrades
    - Tuning
    - Additional observations
- **Maintain** operational model suite
  - The scientific correctness and integrity of operational forecast modeling systems
  - Modify current operational system to adapt to ever-present external changes
- **Transition and Develop** operational numerical forecast models from research to operations
  - Transform & integrate
    - Code
    - Algorithms
    - Techniques
  - Manages and executes transition process including
    - Government technical and system performance review before implementation



EMC location within the funnel

NWS Seamless Suite of Forecast Products Spanning Weather and Climate



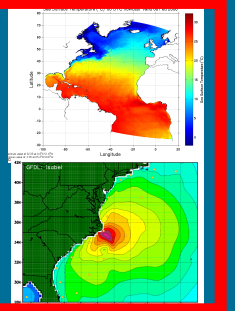
Linkage of Model Systems within Production Suite

CLIMATE CFS
GFS + MOM3/4

Hurricane
GFDL
HWRF (**NMM**)

Coupled

Oceans
HYCOM
WaveWatch III®

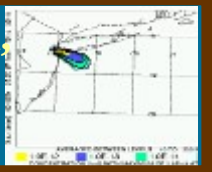


Global Data Assimilation

Global Forecast System

North American Mesoscale
NMM

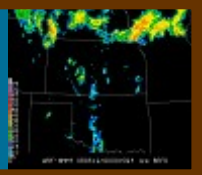
Dispersion, Ash, Smoke & Dust
ARL's HYSPLIT



Regional Data Assimilation

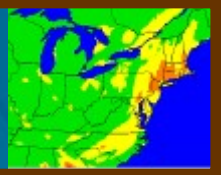
Short-Range Ensemble Forecast
NMM + ARW + ETA + RSM

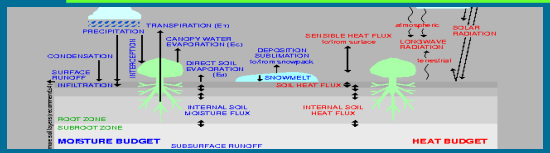
Severe Weather
NMM + ARW



North American Ensemble Forecast System
GFS + Canadian Global +…

Air Quality
NAM + EPA/ ARL's CMAQ

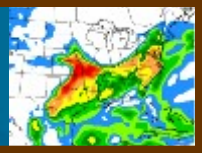




NOAH Land Surface Model

Very Short Range Ensemble Forecasts
Time-Lagged RUC+NAM

Rapid Update for Aviation
GSD's RUC



IBM Power6 p575

- 69.7 Teraflops Linpack
- 156 Power6 32-way Nodes
- 4,992 processors @ 4.7GHz
- 19,712 GB memory
- 320 TB of disk space per system
- 13 PB tape archive

Fairmont, West Virginia

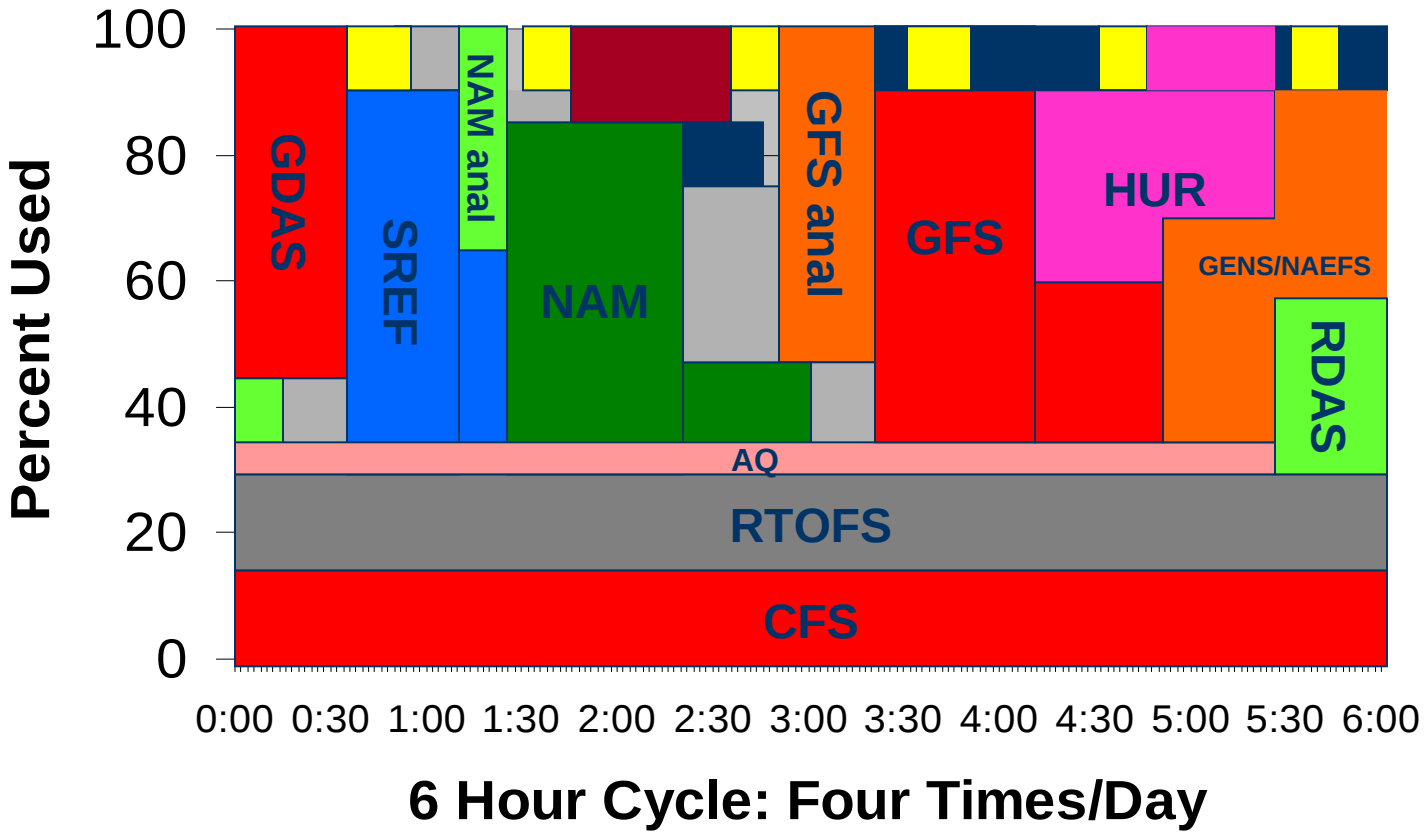
Cirrus— (backup)

Gaithersburg, Maryland

Stratus— (primary)

NCEP Production Suite
Weather, Ocean, Land & Climate Forecast Systems

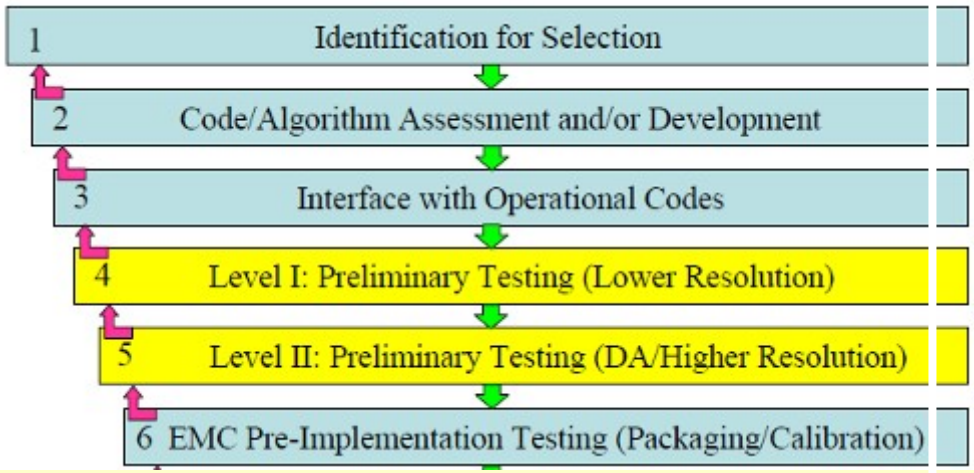
Current - 2009



6 Hour Cycle: Four Times/Day

Implementing upgrades to the NCEP Model Production Suite

R&D and pre-implementation Phase



EMC Change Control Board

- Scientific Integrity
- Product Quality
- EMC Mgmt Approval

- Generate RFC's
- Submit RFC's to NCO

Implementation Phase

- SPA's build NCO parallel from RFC's
- 30-day NCO parallel
  - Test code stability
  - Test dataflow
  - Products to NCEP Centers and EMC code developers
- NCEP Centers
  - Evaluate impact
  - Assessments to NCEP

- 30-day NCO parallel stable
- NCEP centers approve

Briefing to NCEP Director for final approval

Implementation

Running a model in Operations

- Most operational models are run on fixed cycles (4 times a day; 0z 6z 12z 18z)

  These models can be triggered at set times or by the availability of certain necessary files (e.g. Operational WAVEWATCH III® cycle is triggered when a particular GFS file becomes available)

  NOTE: There are exceptions to this (such as the Rapid Update Cycle which is done on an hourly basis and the Ocean Circulation Model which runs on a daily cycle due to computational constraints)

- Models are written in C or Fortran using libraries and utilities that are managed by NCO
- All model executables are wrapped around shell scripts, for

  Proper file management

  Adequate error handling

NOTE: Human interference is limited to monitoring and trouble shooting

Typical model cycle in Production

- A typical model cycle is divided into 4 steps
  1. PREP STEP: This is the step where all the input files necessary for running the model are prepared. (e.g. for WAVEWATCH III this is the step where the wind data from the GFS format would be converted to the internal WW III format using ww3_prep)
  2. FORECAST STEP: This is the step where the raw outputs from the model is generated.
  3. POST STEP: This is the step where output data is converted to suitable form for distribution. (e.g. creating ASCII spectral data files for nesting in SWAN)
  4. PRDGEN STEP: This is the step where model output is distributed (Operational forecast products from NCEP are distributed using AWIPS)

NOTE: Each step is referred to as a JOB

Model Cycle (contd).

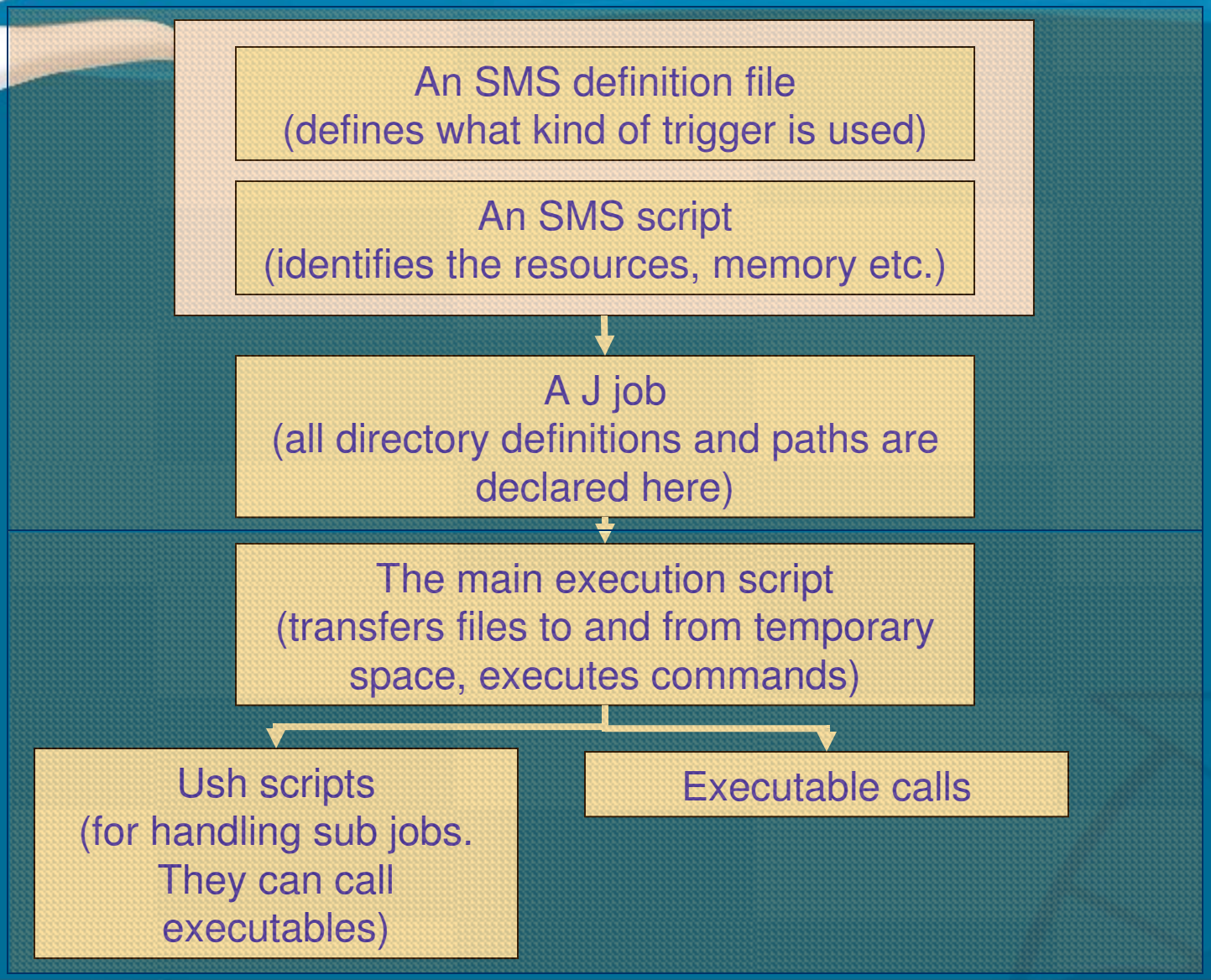
- In addition to cycles run by NCO we run some additional cycles
  1. ARCHIVE STEP: Archive select parts of the forecast (output at buoy locations and collocated altimeter tracks) for later analysis (**IMP**: It is NCO's job to make sure models run on time and results are delivered. It is our job to make sure that what is delivered is not nonsense)
  2. WEB STEP: Developing additional products / images for our personal web page. This has been done as a courtesy to the user community and has been designed as a value added service. (NOTE: This is **NOT** the official forecast. That only comes from the Operational Centers)

Job Flow

- NCO uses the **Supervisor Monitor Scheduler (SMS)** to submit production jobs to run on the CCS. The SMS scripts define computer resources needed to run each job, including
  - trigger (may be a time trigger or activation/completion of another job)
  - memory
  - number of nodes/processors per node
  - parallel vs. serial, shared vs. not shared
  - environment (prod, para, test)
  - Wall clock (maximum runtime)
  - cycle (usually 00Z, 06Z, 12Z, 18Z)

- SMS is freely available from the European Center for Medium-range Weather Forecasts (ECMWF) and is preferred because
  - It provides flexible triggers for starting jobs
  - Has a convenient Graphic User Interface for monitoring job flow

Nuts and bolts of a Job



NCO responsibility

Our deliverables to NCO. Significant portions of the codes are dedicated to error handling

Directory Structure

- There are 3 different types of disk space that are used for executing the jobs

Permanent files, changes are slow →

The save space: This is the area where all the source codes, executable codes, main (and auxiliary) scripts and fixed definition files (such as input and grid) are stored.

Model related I/O, changes every forecast cycle →

The "noscrub" space: This is the space where all necessary generated files (from the jobs) are stored. Files stored here include final model outputs as well as necessary intermediate files (for different jobs).

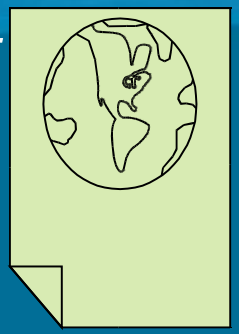
Temporary files →

The scrub space: This is the area where the actual job is run. This space is temporary and all files generated in this space are scrubbed clean after the job

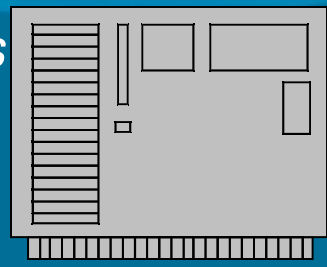
NOTE: The final staging area(s) where data is served to the outside world is separate from this

Data Flow

Forecast info for the user community



Older forecasts moved to tape archives



Save space

/nwprod/scripts (main scripts)
/nwprod/jobs (starting jobs)
/nwprod/exec (executables)
/nwprod/sorc (source codes)
/nwprod/fix (fixed files)

No scrub space

/com/wave/prod/wave.20091201
wave.20091202
wave.20091203
etc.

Files moved to scrub space for executing job

Scrub space

/tmp/JOBID etc

Necessary files moved back to no scrub space after job executes (file names include forecast cycle)

Sample SMS script

- .def file, which defines trigger(s)

```
suite prod00
   family wave00
     family prep
       extern /prod00/gfs00/post/jgfs_post_f12_00
       task jwave_multi_1_prep_00
         trigger /prod00/gfs00/post/jgfs_post_f12_00 ==
             complete
         edit SMSCMD '/sms/smsutils/unixsubmit  %SMSJOB%
                      %SMSJOBOUT% ibmsp'
         edit SMSPASS 'FREE'
       endtask
     endfamily
   endfamily
 endsuite
```

Sample sms script (contd).

- .sms file

```
export MP_SHARED_MEMORY=yes
export MEMORY_AFFINITY=MCM

# EXPORT list here
set -x
export envir=prod
export cyc=00
export job=wave_multi_1_prep_00

# CALL executable job script here

SMSNAME=%SMSNAME% export SMSNAME
SMSNODE=%SMSNODE% export SMSNODE
SMSPASS=%SMSPASS% export SMSPASS
SMS_PROG=%SMS_PROG% export SMS_PROG

SMSBIN=/nwprod/sms/bin export SMSBIN

/nwprod/jobs/JWAVE_MULTI_1_PREP.sms.prod
```

Sample J job (some parts)

```
#!/bin/sh

$SMSBIN/smsinit $LOADL_STEP_ID
export NET=wave
export RUN=wave
export HOMEwave=/nw${envir}
export EXECwave=$HOMEwave/exec
export FIXwave=$HOMEwave/fix
export PARMwave=$HOMEwave/parm
export USHwave=$HOMEwave/ush
export com=/com/${NET}/${envir}
export COMIN=/com/${NET}/${envir}/${RUN}.${PDY}
export COMOUT=/com/${NET}/${envir}/${RUN}.${PDY}
export COMGFS=/com/gfs/prod/gfs.${PDY}
export COMICE=/com/omb/prod
# Execute the script.
/nw${envir}/scripts/exwave_multi_1_prep.sh.sms
```

IMPORTANT!! All directory paths are declared at the top level J job script. No hard wired directories in the main execution script or the auxiliary ush scripts

Coding Standards at NCEP

- Source code should be written in Fortran or C. All source is stored in /nwprod/sorc on the CCS, in a directory of the form

  fortran_sorc.fd or
  c_sorc.cd

- Makefiles should use the IBM compilers for these languages.

- **Code should be written from an operational perspective**

  Source code and scripts must have documentation blocks.
  eases troubleshooting

Coding Standards (contd)

- Executable code within scripts should be wrapped in error checking.

- Error messages should be as descriptive as possible. Failures should not be allowed to propagate downstream of the point where the problem may be detected.

  speeds the time in which NCO personnel can determine why a job failed

  allows job to be rerun (when appropriate) as quickly as possible
  to minimize delays in product delivery

- Executables should not terminate abnormally due to an error which is discoverable/trappable.

  e.g. missing input files should be handled within the script before the executable runs, or within the executable itself

Product Delivery

- The developer works with NCO's **DATAFLOW Team** to determine how/when/where to disseminate model output.
- NCO uses DBNet to disseminate products. Products may be sent to NCEP or NWS ftp servers, AWIPS, etc.
- **Changes in product delivery time or product content must be advertised via a Technical Implementation Notice (TIN).**

Changes to Production

- All changes to the production job suite must be made via a Request For Change (RFC). An RFC includes
  - description of the change being made
  - benefits/risks of the change
  - who is impacted by the change
  - change in resources (disk space, memory, etc.)
  - testing performed prior to submission of RFC
  - names of files being changed
  - implementation instructions
- Any changes to production must be tested by the developer prior to RFC submission. NCO's Senior Production Analyst (SPA) Team then tests each RFC in the appropriate environment prior to scheduling the RFC for implementation.
- major model changes need 30-day parallel test w/ evaluation
- Changes affecting the user community (new models/grids change in delivery products etc.) need to be advertized by a TIN which requires a 30 – 90 day

Take Home message

- All of these “seemingly bureaucratic” processes are essential to ensure that
  - NCO is able to execute the 2 million + job cycles daily and maintain its standard of on time delivery (defined as delivering product within 15 minutes of stated time) 99.96% of the time !
  - Transitions of new code/changes to operations happen in as seamless a fashion as possible
  - There is an effective process to communicate changes to downstream users
  - Trouble shooting can be done by an easily trained operator without requiring too much inside knowledge
- To ensure that we (scientists) transfer an operational code to NCO, the development environment should try to mimic the operational environment as close as possible

The end

End of Lecture 1.3